**《算法设计与分析》实验报告**

# 实验二 动态规划

|  |  |  |  |
| --- | --- | --- | --- |
| **报告书** | | | |
| 姓名 | 吴宇敖 | 指导教师 |  |
| 学号 | 176001752 | 日 期 | 2020.3.26 |
| 班级 | 17计算机软件一班 |  |  |
| **实验内容** | | | |
| **[要求先在OJ上提交通过]**   1. 一家公司购买长钢条，将其切割成短钢条出售，切割本身没有成本，长度为i的短钢条的价格为Pi。那给定一段长度为n的钢条和一个价格表Pi,求钢条的切割方案使得收益Rn最大。[OJ]      1. 求2个字符串的非连续最长公共子序列[OJ] | | | |
| **实验目的** | | | |
| 1）熟练掌握动态规划  2）面对问题时可以将问题进行合理的分析，将问题分解成为若干个小问题，之后再求全部小问题的和。 | | | |
| **实验过程和步骤** | | | |
| **【实验题目】**  给定两个字符串，求解这两个字符串的最长非连续（允许连续或非连续）的公共子序列的长度（Longest Common Sequence）。  比如字符串1：BDCABA；字符串2：ABCBDAB。  则这两个字符串的最长公共子序列长度为4，最长公共子序列是：BCBA  输入要求：  输入2行，每行一个字符串；字符串长度<1000。  输出要求：  输出两个字符串的最长非连续的公共子序列的长度。  **【程序代码及注释】**  #include<iostream>  #include<string>  using namespace std;  int res[1001][1001]={0};//用来记录不同两个长度之间的最长公共子序列  int main()  {  string num1,num2;//定义两个空间来存放两个字符串  getline(cin,num1);  getline(cin,num2);  for(int i=1;i<=num1.size();i++)//将第一个字符串的字符与另一个字符串的字符进行比较  for(int j=1;j<=num2.size();j++)  if(num1[i-1]==num2[j-1])//如果相同那么长度为i和j的最长子序列长度就是长度为i-1和j-1的最长子序列的长度加1  res[i][j]=res[i-1][j-1]+1;  else  res[i][j]=max(res[i-1][j],res[i][j-1]);//如果不相同，那么将会取长度为i和j-1的最长子序列的长度与长度为i-1和j的最长子序列的长度之间最大的值  cout<<res[num1.size()][num2.size()]<<endl;  //system("pause");  return 0;  }  **【实验结果及遇到的问题】**    遇到的问题：  （1）一开始我使用了字符数组进行输入，但是输入一直进行出错。之后我选择了string类型的定义，然后使用getline（）进行输入。  （2）在进行循环时我使用的时1到n，这样可以防止在下面保存结果时的res[i-1][j-1]时，防止超过数组的范围，但是在之后的两个数进行比较时，我忘记将两个字符数改为num1[i-1]==num2[j-1]。  **【算法分析】**  这次实验时寻找到最长子序列，我通过两个循环来进行查找，当两个字符相同时，那么长度为i和j的最长子序列长度就是长度为i-1和j-1的最长子序列的长度加1，如果不同时那么将会取长度为i和j-1的最长子序列的长度与长度为i-1和j的最长子序列的长度之间最大的值。  **【实验题目】**    输入要求：  输入钢条的长度n  输出要求：  输出获得的最大收益  **【程序代码及注释】**  #include<iostream>  using namespace std;  int main()  {  int n,x;  int m[]={0,1,5,8,9,10,17,17,20,24,30};//定义各种长度的价值，0的时候为0  int res[1001]={0};//定义某个长度的最大价值  cin>>n;  for(int i=1;i<=n;i++)//从开始慢慢的往长度n推导  {  x=0;//价值  for(int j=1;j<=10;j++)//进行不同的切割  {  if(i<j)  break;  else  x=max(x,res[i-j]+m[j]);//进行不同的切割后，将切割后的价值进行比较，保留最大价值  }  res[i]=x;  }  cout<<res[n]<<endl;  //system("pause");  return 0;  }  **【实验结果及遇到的问题】**    遇到的问题：由于这题一开始不知道如何去进行求解，经过分析后，只能写出递归的代码，但是由于递归方法发费时间太久，在oj上面通不过，之后将递归给归纳优化为递推  **【算法分析】**  这次实我是从长度1到长度n来进行计算，当满足切割的条件时，将会比较一下切割和不切割之间的价值差。选择价值最大的那一种。 | | | |